ภาษิต ภัทรมูล

SOFT MAIL

ลีก ๆ กับ Visual FoxPro 5 ตอนที่ 2

แนะนำ 2 วิธีการ
สำหรับสร้าง Progress bar
ซึ่งเป็นคอนโทรลในกลุ่ม
ActiveX ของ Visual FoxPro 5

บทความตอนนี้ จะว่าไปดูจะไม่ค่อยมีสลักสำคัญอะไรนัก อย่าเพิ่งมองข้ามครับ เพราะความสำคัญหรือไม่สำคัญบางทีก็วัดไม่ได้กับสิ่งเล็ก ๆ น้อย ๆ ที่เรากำลังทำอยู่ เพราะผลที่ตามมานั้นอาจจะยิ่งใหญ่ก็เป็นได้

ผู้เขียนเคยเกริ่นเมื่อบทที่แล้วว่า Progress bar ซึ่งเป็นคอนโทรลในกลุ่ม ActiveX ของ Visual FoxPro 5 นั้นไม่สามารถเรียกใช้ได้ (เคยลองใช้วิธีต่าง ๆ อยู่เกือบ 2 ชั่วโมง) ความจริงนั้นอาจจะมีทาง แต่ถ้ามันยุ่งยากนักเรามาสร้างเองเลยดีกว่า

เนื้อหา

บทความนี้จะเน้นเนื้อหาใน 2 ส่วน คือ

1. การสร้าง Progress bar โดยการดัดแปลงจาก Controls ต่าง ๆ โดยการใช้ Form Editor

2. การสร้าง Progress bar โดยการเขียนโค้ดเรียกจากคลาสของ Visual FoxPro โดยตรง ซึ่งวิธีนี้อาจดูยุ่งยากซับซ้อน แต่จะเป็นผลดีสำหรับพวกเราที่สนใจจะเขียนโปรแกรมโดยการใช้คลาส หรือจะสร้างคลาสขึ้นมาเอง ซึ่งเมื่ออ่านและทดลองใช้งานแล้วจะเกิดความเข้าใจในเรื่องคลาสมากยิ่งขึ้น ผลที่ได้รับบอกตรง ๆ ว่าคุ้มคุ้มอย่างไรติดตามอ่านได้ครับ

Progress bar

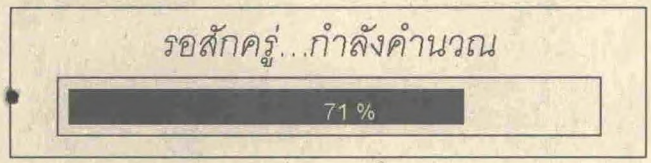
ตามความเข้าใจของผู้เขียน Progress bar ก็คือตัวบอกข่าวสารแก่ผู้ใช้ ว่าขณะนี้โปรแกรมของเรากำลังทำอะไรอยู่ และดำเนินการไปได้เท่าไร โดยแสดงเป็นแถบสีแทนการทำงาน หรืออาจมีส่วนของตัวเลขเปอร์เซ็นต์เพิ่มขึ้นมา

รูปแบบของ Progress bar จึงสามารถมีหลากหลายรูปแบบ โดยไม่จำเป็นต้องยึดติดตายตัวกับแบบใดแบบหนึ่ง แม้แต่ตัวที่วินโดวส์ใช้อยู่ก็ตาม

ประโยชน์สำหรับโปรแกรมฐานข้อมูลคือ บางงานที่ต้องใช้เวลานาน เช่น การคำนวณ หรือประมวลผลข้อมูลจำนวนมาก บางครั้งเป็นหมื่นถึงแสนเรคอร์ด ในระหว่างที่ทำงาน ข่าวสารที่จะบอกต่อผู้ใช้เป็นส่วนสำคัญอย่างยิ่ง เพราะการที่จะปล่อยให้โปรแกรม (เหมือนกับ) หยุดไปขณะทำงานหลายนาที ผู้ใช้ย่อมเกิดความสงสัย บางทีอาจคิดว่าระบบแฮงค์ไปแล้วก็ได้ ใครจะบอกได้ว่าเมื่อผู้ใช้ตัดสินใจกดรีเซ็ตแล้ว ระบบฐานข้อมูลที่ทำงานอยู่จะมีสภาพเป็นอย่างไร? อาจเป็นไปได้ว่าอินเด็กซ์ไฟล์เสีย หรือหนักหนาสาหัสถึงขนาดไม่สามารถเรียกใช้แฟ้มข้อมูลที่เสียหายได้อีกเลย

รูปแบบพื้นฐานของ Progress bar

รูปแบบ Progress bar ที่จะใช้ ขอให้เป็นอย่างรูปที่ 1



รูปที่ 1

ซึ่งจะมีองค์ประกอบพื้นฐาน ทั้งข่าวสาร แถบสีแสดงการทำงาน และอัตราเปอร์เซ็นต์ ลักษณะที่สำคัญอีกอย่างของ Progress bar คือ ปรากฏตัวเมื่อทำงาน และทำลายตัวเองโดยอัตโนมัติเมื่อทำงานเสร็จ

การสร้าง Progress bar โดยใช้ Form Editor

- เปิดโปรแกรม Visual FoxPro เข้าสู่ Project Manager คลิกที่ Documents เลือก Forms เพื่อสร้าง (New) ฟอร์มขึ้นมาใหม่
- เลือกคอนโทรลมาปะโดยคุณสมบัติของคอนโทรลที่สำคัญมีดังในตารางที่ 1 และรูปแบบ (ดูในรูปที่ 2)

ตารางที่ 1

ชื่อ	Properties	Assign	Events
Form1	Caption	กำลังทำงาน!	
	Autocenter	.T.	
	Clipcontrol	.F.	
	Showwindow	1	
			Activate
			Init
Label1	Autosize	.T.	
	Backcolor	192,192,192	
	Back Style	0	
	Caption	0%	
	Forcolor	255,255,0	
Label2	Caption	รอสักครู่..กำลังคำนวณ	
	Alignment	2	
Shape1	Fillcolor	0,64,0	
	Fillstyle	0	
	Width	400	
Shape2	Backcolor	0,0,255	
	Bordercolor	255,255,0	

รูปที่ 2 (Form1)

ฟอร์มตัวอย่างอาจมีลวดลายแปลกตา ตรงนี้ไม่มีอะไรมาก เพราะผู้เขียนแทรกภาพที่ทำจาก Adobe Photoshop เข้าไปเป็นฉากหลังเท่านั้นเอง ความสวยงามนั้นกินไม่ได้แต่เท่ครับ

สำคัญที่ Events

เมื่อตกแต่งฟอร์มเสร็จ ส่วนที่สำคัญที่จะทำให้ฟอร์มทำงานในรูปแบบที่เราต้องการแต่ต้นคือ ปรากฏขึ้นมาทำงานในทันที และทำลายตัวเองเมื่อเสร็จงาน นั่นคือเราต้องเขียน Events ของฟอร์ม

Events ที่ใช้เราต้องไปกำหนดที่ Properties ของ Form นั่นคือ Event init (ลิสติงที่ 1) และ Event Activate (ลิสติงที่ 2)

ลิสติงที่ 1

```
01 Public A
02 A=0
```

ลิสติงที่ 2

```
01 If a=0
02 ccc=10000
03 for n=1 to ccc
04 thisform.shape1.width= (n/ccc) *400
05 thisform.label1.caption=allt (str (int
   (thisform.shape1.width/4) ) ) +' %'
06 thisform.refresh
07 endf
08 endi
09 thisform.deactivate
10 thisform.release
11 cancel
```

อธิบายเพิ่มเติม

ในลิสติงที่ 1 เป็นการสร้างตัวแปรสาธารณะคือ A กำหนดให้ A=0 ที่ต้องกำหนดเพราะ Event Activate นั้นจะทำงานซ้ำแล้วซ้ำเล่าถ้าไม่มีตัวเช็ค ในที่นี้ Event Activate จะทำงานก็ต่อเมื่อ A=0 เมื่อทำงานเสร็จจะบวกค่าใน A อีก 1 (บรรทัดที่ 01 และ 09 ในลิสติงที่ 2) เมื่อวนรอบต่อไป A ไม่เท่ากับ 0 ก็จะออกจากฟอร์มโดยการทำลายตัวเอง

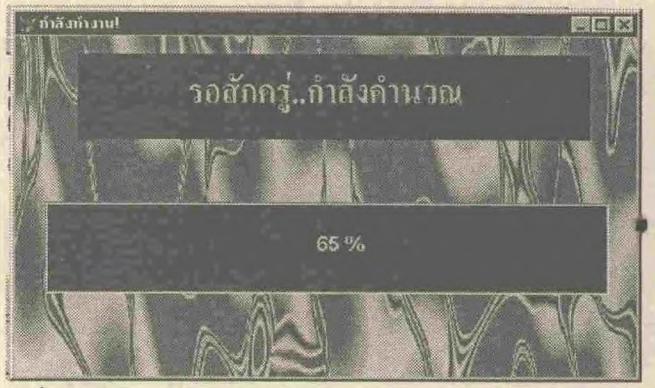
ในลิสติงที่ 2 จะเป็นการทำงาน โดย CCC คือจำนวนงานทั้งหมดที่ต้องปฏิบัติ จากนั้นจึงเป็นการลูปโดยใช้ประโยค For...Endfor (03-07) โดยตรวจเช็คค่าของ n ซึ่งจะเพิ่มขึ้นเรื่อย ๆ ตามการลูป ค่าของ n จะมีผลต่อความกว้าง (properties Width) ของ Shape1 ซึ่งกำหนดค่าสูงสุดไว้เป็น 400 คือ 100% ของงานที่บรรลุผล ลองรันฟอร์มได้เลยครับ (ดูรูปที่ 3)

เราอาจนำไปดัดแปลงใช้ในการประมวลผลของแฟ้มข้อมูลนั่นคือ (ดูลิสติงที่ 3)

1. เลือกพื้นที่ดำเนินการ (01-05)
2. นับจำนวนงานที่ทำทั้งหมดใส่ใน CCC (06)
3. แทรกการทำงานในแฟ้มข้อมูล (12-18)
4. Skip ไปงานต่อไป (19-20)

ลิสติงที่ 3

```
01 If a=0
02 Use cust in 1
03 Use pay in 2
04 Use work in 3 inde work orde tag idcust
05 Sele 1
06 Calc cnt () to ccc
07 Go top
08 for n=1 to ccc
09 stor idcust to idxcust
10 thisform.shape1.width= (n/ccc) *400
11 thisform.label1.caption=allt (str (int
   (thisform.shape1.width/4) ) ) +' %'
12 sele 2
13 calc cnt () to sumw for idcust=idxcust
14 sele 3
15 seek idcust
16 if foun ()
17 repl sumwork with sumw
18 endi
19 sele 1
20 skip
21 thisform.refresh
22 endf
23 endi
24 a=a+1
25 thisform.deactivate
26 thisform.release
27 Cancel
```



รูปที่ 3

ปัญหาที่พบจากวิธีนี้

ที่เห็นชัด ๆ คือ ค่อนข้างเปลืองเนื้อที่ เพราะจะโชว์ Progress bar ทีก็ต้องทำฟอร์มอีกฟอร์มหนึ่ง โปรแกรมที่เราคอมไพล์แล้วจะมีขนาดใหญ่ขึ้น ยิ่งถ้าใช้ Progress bar สัก 10-20 ตัว ดูแล้วน่าใจหายครับ ตามความเป็นจริงต่อให้เราใช้ Progress bar ในกลุ่มคอนโทรล ActiveX ได้ ความสิ้นเปลืองก็ยังคงมีอยู่ดี วิธีแก้คือ ใช้ให้น้อย กรณีที่จำเป็นเท่านั้น หรือจะใช้วิธี Save ฟอร์มนี้ไว้เป็นคลาส แล้วเรียกใช้โดย Override เฉพาะใน Procedure activate ที่เป็นส่วนประมวลผลของฟอร์ม ซึ่งจะกล่าวต่อไปนี้

การสร้าง Progress bar โดยการใช้คลาสของ Visual FoxPro

ก่อนอื่นจะขอกล่าวถึงการ Override ใน Visual FoxPro เสียหน่อย ตามลักษณะโปรแกรมที่ใช้หลัก OOP ย่อมจะต้องมีการสืบสกุลประดุจเดียวการมีลูกมีหลาน ซึ่งลูกหลานก็ย่อมมีลักษณะทางกรรมพันธุ์และอุปนิสัยคล้ายกับบรรพบุรุษ (อาจไม่เหมือนร้อยเปอร์เซ็นต์ เพราะขึ้นอยู่กับยีนเด่น-ยีนด้อยของของพ่อแม่ด้วย) แต่ทางด้านคอมพิวเตอร์แล้วลูกหลานจะออกมาเหมือนบรรพบุรุษทุกประการ ยกเว้นว่าเราจะไปแทรกแซงในบางจุดที่ทำให้ลักษณะการทำงานนั้นเปลี่ยนไป การแทรกแซงซึ่งมีอำนาจเหนือกว่าลักษณะจากบรรพบุรุษเดิมเรียกว่า Override ครับ

สำหรับ Visual FoxPro การ Override จะทำได้ง่ายมาก กล่าวคือ ถ้าเราระบุ Properties หรือ Method ที่เป็นชื่อเดียวกับที่มีในคลาสบรรพบุรุษ Visual FoxPro จะยอมรับโดยปริยายว่าเป็นการ Override และจะเปลี่ยนการทำงานให้แบบอัตโนมัติ

ใน Progress bar เราจะ Override ในจุดเดียวคือใน Procedure activate ซึ่งเป็นจุดที่เราต้องเปลี่ยนแปลงการประมวลผลโดยการเรียกใช้แฟ้มข้อมูล ซึ่งมีชื่อแตกต่างกันตามความต้องการ การเขียนโค้ดให้ดูตามลิสติงที่ 3

การสร้างคลาส

เปิด Form Editor ของ Form ในตัวอย่าง คลิกที่เมนู File เลือก menu ย่อย Save as class จะมีหน้าต่างขึ้นมาให้เติมชื่อของคลาส ในตัวอย่างจะใช้ชื่อว่า jclass เก็บใน D:\jclass.vcx เมื่อ Save เป็นที่เรียบร้อยแล้ว ปิด Form Editor กลับมาที่ Project manager คลิกที่ตัวเลือก Class เลือก Add นำ Class ที่ชื่อ Jclass จากไดรฟ์ D: มาใช้ในโปรเจ็กต์ของเรา

ส่วนโค้ดของการเรียกใช้ดูในลิสติงที่ 4

ลิสติงที่ 4

```
*progress.prg
01 set defa to d:\
02 set classlib to jclass addi
03 progress = CREATEOBJ ('jexam')
04 progress.show
05 read event
06 clea event
07 defi class jexam as jclass
08 proc activate
09 if a=0
10 set talk off
11 set curs off
12 n=1
13 ccc=10000
14 for n=1 to ccc
15 thisform.shape1.width= (n/ccc) *400
16 thisform.label1.caption=allt (str (int
   (thisform.shape1.width/4) ) ) +' %'
17 thisform.refresh
18 endf
19 endi
20 a=a+1
21 thisform.deactivate
22 thisform.release
23 cancel
24 endproc
25 enddefine
```

อธิบายเพิ่มเติม

1. บรรทัดที่ 01 ตั้ง path ไปที่ d:\

2. บรรทัดที่ 02 เรียกใช้คลาส jclass ที่เก็บใน d:\

3. บรรทัดที่ 03-05 เรียกใช้และโชว์ Class ที่ชื่อ Jexam ที่สร้างในบรรทัดที่ 07-25

4. บรรทัดที่ 07 สร้าง Class ชื่อ jexam จากคลาสบรรพบุรุษ jclass ใน jclass.vcx

5. บรรทัดที่ 08-24 คือ procedure activate ที่เรา Override อีเวนท์ Form.activate ของเดิม ซึ่งตรงนี้เราสามารถปรับเปลี่ยนได้ตามความต้องการใช้งาน

ข้อดี

ที่เห็นชัด ๆ คือประหยัดพื้นที่เก็บฟอร์มได้มากครับ เพราะเราใช้แค่ฟอร์มเดียว แล้วเปลี่ยนแค่โค้ดในการประมวลผลเท่านั้น ผลพวงที่ตามมาคือความเร็ว ซึ่งเป็นยอดปรารถนาของหลาย ๆ คน ทั้งนี้เพราะโปรแกรมที่คอมไพล์แล้วจะมีขนาดกะทัดรัด ยุคไอเอ็มเอฟเราจะต้องประหยัดครับ

Code ล้วน ๆ

พวกเราที่ชอบเขียนโค้ด Define Class เป็นชีวิตจิตใจ ไม่อยากมาเกลือกกลั้วกับ Form Editor ผู้เขียนก็เตรียมมาให้เหมือนกัน ตรงนี้ไม่ต้องสร้างมาฟอร์มให้เสียความรู้สึก เขียนโค้ดแล้วรันได้เลย! ดูลิสติงที่ 5 (เป็นของแถม)

ลิสติงที่ 5

```
01 *เรียกใช้ Class ที่สร้าง
02 progressbar=createobject ("prog_exam")
03 progressbar.show
04 read event
05 CLEA EVEN
06 clea class prog_exam
07 *สร้าง Class
08 defi class prog_exam as form
09 caption="กำลังทำงาน!"
10 autocenter=.t.
11 clipcontrols=.f.
12 height=271
13 left=56
14 top=11
15 width=491
16 showwindow=1
17 *เพิ่ม Shape2
18 add object shape2 as shape ;
19 with backcolor=rgb (0,0,255) ,;
20 bordercolor=rgb (255,255,0) ,;
21 height=73,;
22 left=24,;
23 top=132,;
24 width=445
25 ***เพิ่ม Shape1
26 add object shape1 as shape ;
27 with fillcolor=rgb (0,64,0) ,;
28 fillstyle=0,;
29 height=49,;
30 left=48,;
31 top=144,;
32 width=400
33 ****เพิ่ม    label1
34 add object label1 as label ;
35 with autosize=.t.,;
36 backcolor=rgb (192,192,192) ,;
37 backstyle=0,;
38 caption="0%",;
39 fontname="Arial",;
40 fontsize=12,;
41 forecolor=rgb (255,255,0) ,;
42 height=22,;
43 left=240,;
44 top=156,;
45 width=31
46 ***เพิ่ม label2
47 add object label2 as label ;
48 with height=72.,;
49 left=48,;
50 top=12,;
51 width=408,;
```

→